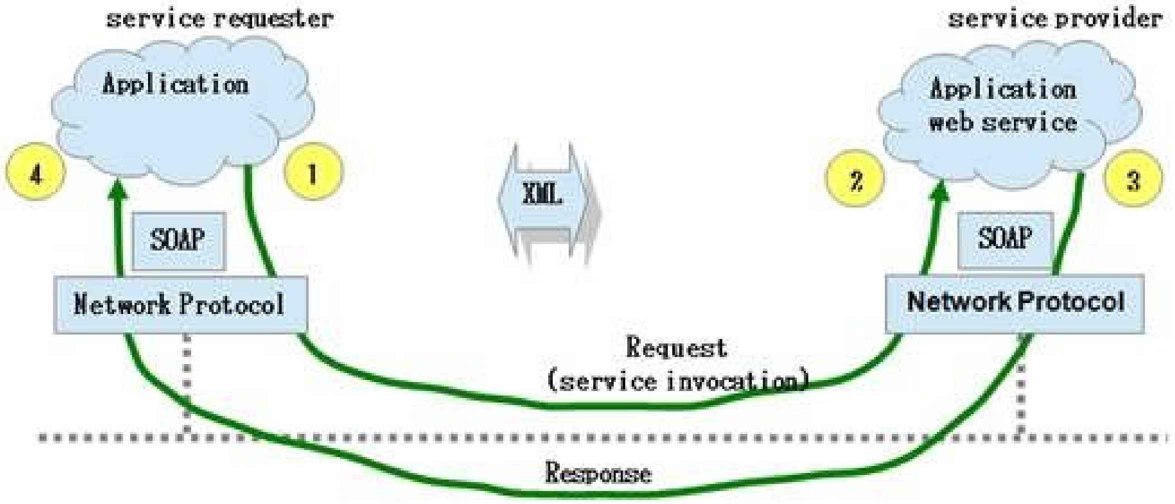
1. Design and Implementation:



The WDSL/SOAP Web Service designed with the reference of the image above. There is a client and a server which has a SOAP web service layer written in JAVA. The server is TomCat ran using SpringBoot and Java Web Servlet. The project structure is designed and implemented using maven project structure. The calculator operations are implemented using endpoints in both the approaches which are annotated with @WebService. The interface and implementation are added into the same endpoint class and not separated.

In Top-Down Approach, xsd files were written and used to generate a wsdl file using a plugin. This wsdl file was then used to prepare the data models required for the web services. The design involved just one response type <xs:element name="getCalcResponse"> and a request type <xs:element name="getCalcRequest">.

In Bottom-Up Approach required models were created in CalcModel class to inculcate the calculator operations. The required functionalities from this model were used in the Web Service Endpoint implementation. Requests were added as xml files and the project was tested by hitting the web service point in the server using curl to simulate an application client.

Propose a solution to make the Calculator stateful
(do not implement
just describe the solution: text + diagram)

Normally, a JAX-WS Web service by default is stateless, i.e local variables and values that are set in the web service object are not saved from one invocation to the next. For cases like where a client wants to save data on the service during any invocation and then use that data during subsequent invocation, stateful implementation is necessary. In a calculator application like this, multiple variables can be added upon repeated invocation after one another to do a commutative operation using the calculation operation. The possible solution is to use **HTTP Sessions** to maintain state across Web Service invocations. When statefulness is not required by the user, the session can be destroyed to cut the state in client and server alike. A web client can persist the session token using the functionality of cookies.